## Overview

You will write a Java application to build and search a B-Tree.

## Data Structures

You will need to build a Node object to be the base class for the following two node types:

RootNode - holds a start of the range, end of the range and some number of Nodes (can be RootNodes or LeafNodes).

LeafNode - holds any number of integer values.

There should be a single RootNode that is the "top" of the tree. Its range should encompass the range of the whole tree.

## Searching

If the current node is a RootNode and the number that we are looking for is between the start and end of the range, follow the nodes that are descendents of this RootNode. If the current node is a LeafNode, check each value of the leaf node to see if it matches the value that we are searching for.

## Rules:

1) No iterators, loops or Java functions that iterate for you. All iteration must happen by recursion.

2) No global or static variables. The class that does the searching should not have any members.

3) Please make constructors:

LeafNode(Collection<int> values)

RootNode(int min, int max, Collection<Node> nodes)

Remember to work independently. This is not a large (lines of code) assignment. It is an assignment that is designed to make you think in a different way.

***MAKE SURE TO TEST YOUR CODE!*** Make a B-Tree and search for numbers that are in it and numbers that are not in it.

Note – this is not quite a typical B-Tree; the assignment is a slightly simplified version of an actual B-Tree.

Example B-Tree diagram:

A close up of a map

Description generated with high confidence